

# 리듬게임 DDR

**학과** Median&  
Entertainment

**학번** 20191108

**이름** 신규찬

# 왜?

# 갑자기 리듬게임 인가?

## 01

### 큰 이유



해적 게임 매핑을 좀비 게임을 I wanted to refer to the zombie  
참고 하려 했는데 맵 매핑이 너 game for pirate game mapping,  
무너무 어렵고 코딩이 힘들어서 but it's so hard to code

## 02

### 작은 이유

얼마 전 게임방을 갔는데  
추억의 펌프를 하다가  
DDR을 코딩으로 만들어 보자!  
라는 생각을 했습니다.

Recently, I went to a game  
room. I was playing a pump  
game  
I thought, "Let's code DDR!"

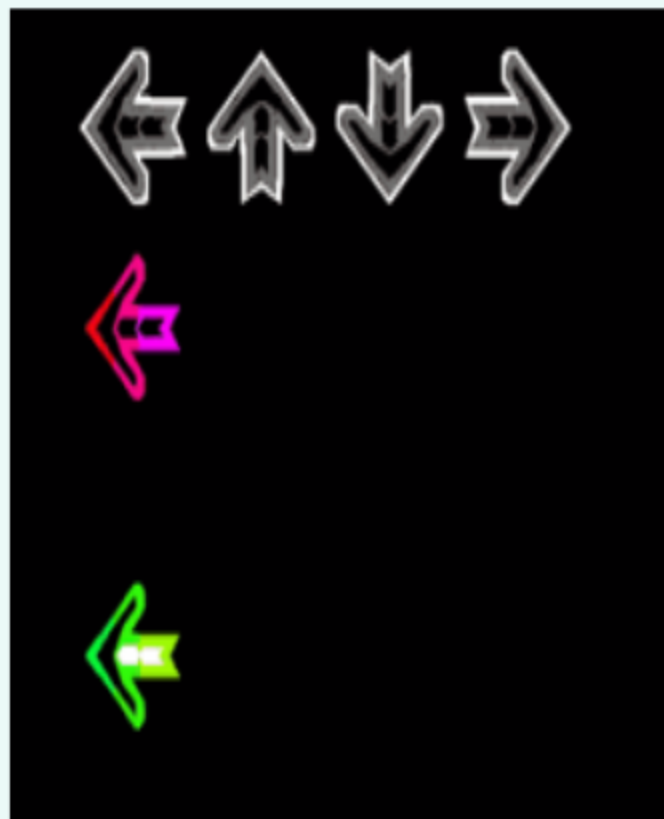
## CHAPTER 02

# 현재

# 진행 상황

### 우선 구현하려는 게임

위에서 음악과 함께 내려오는 방향키 틀을 타이밍에 맞춰서 눌렀을 경우	The direction keys that come down with the music
아주 정확하면 최고점수,	If you press it according to the timing
약간 부정확하면 중간점수	If it's very accurate, the highest score,
누르지 못했을 경우 목숨이 하나 줄어들게 하고 싶습니다	If it's a bit inaccurate, the middle score
	If I can't press it, I want to reduce my life



# 01

```
class Direction(object):
    def __init__(self):
        self.pos = None
        self.direction = 0
        self.image = pygame.image.load(f"리듬게임/direction.png")
        self.image = pygame.transform.scale(self.image, (60, 60))
        self.rotated_image = pygame.transform.rotate(self.image, 0)
        self.y = -1
        self.x = int(SCREEN_WIDTH*0.75)-(self.image.get_width()/2)

    def rotate(self, direction=0):
        self.direction = direction
        self.rotated_image = pygame.transform.rotate(
            self.image, 90*self.direction)

    def draw(self):
        if self.y >= SCREEN_HEIGHT:
            self.y = -1
            return True
        elif self.y == -1:
            return False
        else:
            self.y += 1
            self.pos = screen.blit(self.rotated_image, (self.x, self.y))
            return False
```

## 코드 진행상황

- `self.pos`: 화살표의 현재 위치를 나타내는 변수를 초기화
- `self.direction`: 화살표의 방향을 나타내는 변수를 초기화
- `self.image`: 화살표 이미지를 로드하고 크기를 조정
- `self.rotated\_image`: 화살표 이미지를 회전시킨 이미지를 초기화
- `self.y`: 화살표의 Y 좌표를 초기화합니다. '-1'로 초기화
- `self.x`: 화살표의 X 좌표를 초기화

`def rotate(self, direction=0):`

: 화살표 이미지를 회전시키는 역할

. `direction` 매개변수를 받아 화살표를 해당 방향으로 회전

`def draw(self):`: 이 메서드는 화살표를 화면에 구현

화살표가 화면 아래로 이동하며 그려지고, 화면을 벗어날 경우 다시 화면 위로 이동하도록 함.

## 02

```
def drawicon():
    global start_ticks, chance

    if chance <= 0:
        return

    elapsed_time = (pygame.time.get_ticks() - start_ticks)
    if elapsed_time > 400:
        start_ticks = pygame.time.get_ticks()
        for direc in Directions:
            if direc.y == -1:
                direc.y = 0
                direc.rotate(direction=random.randint(0, 3))
                break

    for direc in Directions:
        if direc.draw():
            chance -= 1
```

## 코드 진행상황 2

방향 화살표 아이콘을 그리고 이동시키는 역할을 하는 'drawicon()' 함수

1. 'global start\_ticks, chance': 이 부분은 함수 내에서 전역 변수인 'start\_ticks'와 'chance'를 수정할 것 이 변수들은 함수 외부에서 선언되었으며 함수 내부에서도 사용
2. 'if chance <= 0:': 'chance' 변수가 0 이하인 경우, 즉 플레이어의 기회가 모두 소진되었을 때, 함수가 아무 작업도 수행하지 않고 종료

3. 'elapsed\_time = (pygame.time.get\_ticks() - start\_ticks)': 게임이 시작된 후 경과한 시간을 계산

4. 'if elapsed\_time > 400:': 경과한 시간이 400 밀리초보다 큰 경우 아래의 코드 블록 실행

- 'start\_ticks = pygame.time.get\_ticks()': 'start\_ticks' 변수를 현재 게임 시간으로 업데이트

- 'for direc in Directions:': 'Directions' 리스트에 있는 모든 화살표 객체에 대해 반복

- 'if direc.y == -1:': 화살표 객체의 'y' 속성이 -1인 경우 (화면 위에 숨겨진 상태인 경우) 아래의 작업을 수행

- 'direc.y = 0': 화살표의 'y' 값을 0으로 변경

- 'direc.rotate(direction=random.randint(0, 3))': 화살표 랜덤한 방향으로 회전.

5. 'for direc in Directions:': 'Directions' 리스트에 있는 화살표 객체에 대해 반복

- 'if direc.draw()': 화살표 객체의 'draw()' 메서드를 호출하여 화살표를 화면에 표시 만약 해당 화살표가 화면을 벗어나서 더 이상 그려지지 않는 상태라면, 'chance' 변수를 감소 -> 플레이어의 남은 기회를 나타내는 변수를 감소

목표

# 점수 환산 시스템 구현

타이밍에 맞춰서 방향키를 누르는 것 구현

타이밍이 더 잘 맞은 경우, 더 높은 점수를

타이밍이 부정확 하면 낮은 점수를

타이밍을 맞추지 못하면 목숨을 1 차감시키는 시스템

우선 화살표를 누를 박스를 만들고

그 박스에 화살표가 닿았을 시에 점수 변경이 일어나는 시스템을 구현할 예정